This listing of claims will replace all prior versions, and listings, of claims in the application:

The Status of the Claims

1. (Currently Amended) A method of preserving exceptions in code reordering, the method comprising:

receiving a plurality of software instructions including a software instruction at a first location within the plurality of software instructions;

determining ~~if~~that the software instruction is an excepting instruction;

inserting a control speculative version of the software instruction at a second location within the plurality of software instructions ~~if~~in response to determining that the software instruction is an excepting instruction;

replacing the software instruction at the first location with a check instruction at the first location; and

generating a recovery block which branches from the check instruction, the recovery block including a copy of the software instruction.

2. (Original) A method as defined in claim 1, wherein the software instruction comprises a load instruction.

3. (Original) A method as defined in claim 1, wherein the second location is positioned for instruction execution earlier than the first location is positioned for instruction execution.

4. (Original) A method as defined in claim 1, wherein inserting a control speculative version of the software instruction at the second location comprises inserting a control speculative version of a load instruction.

5. (Currently Amended) A method of preserving exceptions in code reordering, the method comprising:

receiving a plurality of instructions including a first instruction;

determining if the first instruction is an excepting instruction;

determining if the first instruction is to be moved upward across a check instruction;

determining a second instruction in the plurality of instructions that computes a previous value of a target register associated with the first instruction ~~if~~when the first instruction is not an excepting instruction and the first instruction is to be moved upward across a check instruction;

determining if a source operand associated with the second instruction is available at the check instruction;

inserting a third instruction into the plurality of instructions to save the value of the target register if the source operand associated with the second instruction is not available at the check instruction; and

inserting a fourth instruction into a recovery block to restore the value of the target register.

6. (Original) A method as defined in claim 5, further comprising inserting a copy of the second instruction into the recovery block if the source operand associated with

the second instruction is available at the check instruction.

7. (Original) A method as defined in claim 6, wherein inserting a copy of the second instruction into the recovery block comprises inserting a copy of the second instruction into the recovery block ahead of a copy of the excepting instruction.

8. (Currently Amended) A method of preserving exceptions in code reordering, the method comprising:

receiving a plurality of instructions including a first instruction;

determining ~~if~~that the first instruction is an excepting instruction;

determining if the first instruction is to be moved upward across a check instruction;

determining if the first instruction is to be moved downward across the check instruction; and

inserting a copy of the first instruction into a recovery block ~~if~~in response to determining that (i) the first instruction is not an excepting instruction, (ii) the first instruction is not to be moved upward across a check instruction, and (iii) the first instruction is to be moved downward across a check instruction.

9. (Original) A method as defined in claim 8, wherein inserting a copy of the first instruction into the recovery block comprises inserting a copy of the first instruction into the recovery block ahead of a copy of the excepting instruction.

10. (Currently Amended) An apparatus for preserving precise exceptions in code reordering, the apparatus comprising:

a processor to execute a plurality of software instructions;

a memory device operatively coupled to the processor to store the plurality of software instructions;

a control speculation module operatively coupled to the processor, the control speculation module being structured to insert a control speculative version of a software instruction into the plurality of software instructions in response to a determination that the software instruction is an excepting instruction; and

an exception handler operatively coupled to the processor, the exception handler being structured to handle an exception associated with the control speculative version of the software instruction.

11. (Original) An apparatus as defined in claim 10, wherein the control speculation module inserts a check instruction into the plurality of software instructions.

12. (Original) An apparatus as defined in claim 11, wherein the control speculation module generates a recovery block which branches from the check instruction.

13. (Currently Amended) An apparatus for preserving precise exceptions in code reordering, the apparatus comprising a computing device structured to:

receive a plurality of software instructions including a software instruction at a first location within the plurality of software instructions;

determine ~~if~~that the software instruction is an excepting instruction;

insert a control speculative version of the software instruction at a second location within the plurality of software instructions ~~if~~in response to determining that the software instruction is an excepting instruction;

replace the software instruction at the first location with a check instruction at the first location; and

generate a recovery block which branches from the check instruction, the recovery block including a copy of the software instruction.

14. (Original) An apparatus as defined in claim 13, wherein the software instruction comprises a load instruction.

15. (Original) An apparatus as defined in claim 13, wherein the second location is positioned for instruction execution earlier than the first location is positioned for instruction execution.

16. (Original) An apparatus as defined in claim 13, wherein inserting a control speculative version of the software instruction at the second location comprises inserting a control speculative version of a load instruction.

17. (Currently Amended) A machine readable medium structured to cause a machine to:

receive a plurality of software instructions including a software instruction at a first location within the plurality of software instructions;

determine ~~if~~that the software instruction is an excepting instruction;

insert a control speculative version of the software instruction at a second location within the plurality of software instructions ~~if~~in response to determining that the software instruction is an excepting instruction;

replace the software instruction at the first location with a check instruction at the first location; and

generate a recovery block which branches from the check instruction, the recovery block including a copy of the software instruction.

18. (Original) A machine readable medium as defined in claim 17, wherein the software instruction comprises a load instruction and the second location is positioned for instruction execution earlier than the first location is positioned for instruction execution.

19. (Original) A machine readable medium as defined in claim 17, wherein a control speculative version of the software instruction at the second location comprises inserting a control speculative version of a load instruction

20. (Currently Amended) A machine readable medium structured to cause a machine to:

receive a plurality of instructions including a first instruction;

determine if the first instruction is an excepting instruction;

determine if the first instruction is to be moved upward across a check instruction;

determine a second instruction in the plurality of instructions that computes a previous value of a target register associated with the first instruction ~~if~~when the first

instruction is not an excepting instruction and the first instruction is to be moved upward across a check instruction;

determine if a source operand associated with the second instruction is available at the check instruction;

insert a third instruction into the plurality of instructions to save the value of the target register if the source operand associated with the second instruction is not available at the check instruction; and

insert a fourth instruction into a recovery block to restore the value of the target register.

21. (Original) A machine readable medium as defined in claim 20, wherein a copy of the second instruction is inserted in the recovery block if the source operand associated with the second instruction is available at the check instruction.

22. (Original) A machine readable medium as defined in claim 21, wherein a copy of the second instruction inserted in the recovery block comprises a copy of the second instruction inserted into the recovery block ahead of a copy of the excepting instruction.

23. (Currently Amended) A machine readable medium structured to cause a machine to:

receive a plurality of instructions including a first instruction;

determine if the first instruction is an excepting instruction;

determine if the first instruction is to be moved upward across a check instruction;

determine if the first instruction is to be moved downward across the check instruction; and

insert a copy of the first instruction into a recovery block ~~if~~in response to determining that (i) the first instruction is not an excepting instruction, (ii) the first instruction is not to be moved upward across a check instruction, and (iii) the first instruction is to be moved downward across a check instruction.

24. (Original) A machine readable medium as defined in claim 23, wherein a copy of the first instruction inserted into the recovery block comprises a copy of the first instruction inserted into the recovery block ahead of an excepting instruction.